Royal Institute of Technology

DD2434 DGM (TREES), HMMS

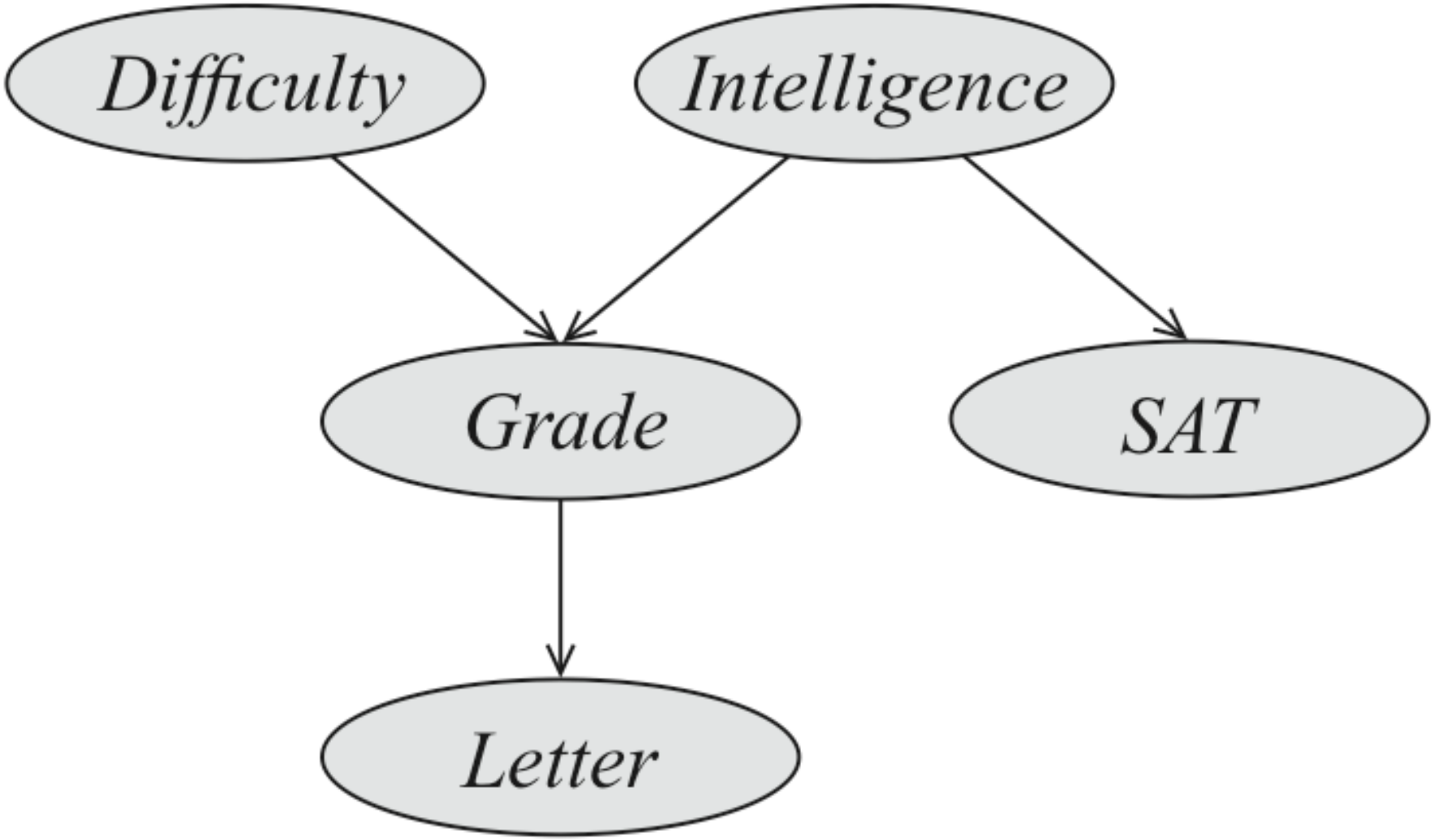
LAST LECTURE?

- ML
- Bayesian (Posterior)

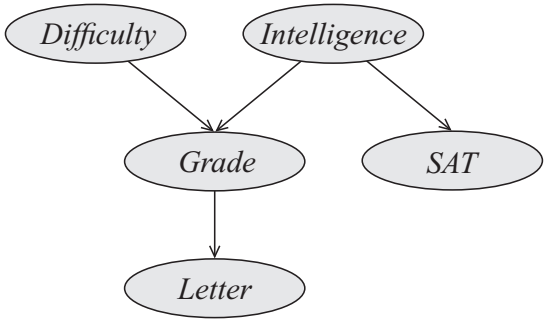
LAST LECTURE

- MLE for categorical
- DGM
- Basic definitions
- Examples
- Learning parameters - given complete data
- Illustrating a known model

STUDENT EXAMPLE



FACTORIZATION - AN EXAMPLE



$$L(\boldsymbol{\theta};\mathcal{D}) = p(D=(0,1,1,1,1)|\boldsymbol{\theta}_D)$$

"Column wise"

$$p(I=(1,1,1,0,1)|\boldsymbol{\theta}_I)$$
$$p(S=(1,1,0,0,0)|I=(1,1,1,0,1),\boldsymbol{\theta}_S)$$
$$p(G=(1,0,0,0,0)|D=(0,1,1,1,1),I=(1,1,1,0,1),\boldsymbol{\theta}_G)$$
$$p(L=(1,0,1,0,1)|G=(1,0,0,0,0),\boldsymbol{\theta}_L)$$

D	I	S	G	L
0	1	1	1	1
1	1	1	0	0
1	1	0	0	1
1	0	0	0	0
1	1	0	0	1

$$N_{v\mathbf{c}k} = \sum_{n=1}^{N} I(x_{nv}=k, x_{n,\mathrm{pa}(v)}=\mathbf{c})$$

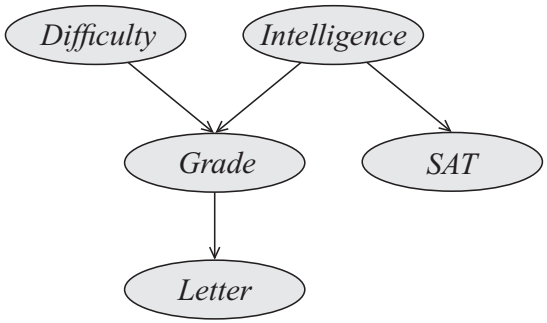
$$N_{v\mathbf{c}} = \sum_{n=1}^{N} I(x_{n,\mathrm{pa}(v)}=\mathbf{c})$$

$$N_{L\langle 0\rangle 0} = 2 \qquad N_{L\langle 0\rangle 1} = 2$$

MLE FOR CAT CPDS

★ Each $P(\mathcal{D}_v|\boldsymbol{\theta}_v)$, i.e., here each $\boldsymbol{\theta}_{v\mathbf{c}}$

can be maximized independently



★ So, MLE is

$$\boldsymbol{\theta}_{v\mathbf{c}k} = N_{v\mathbf{c}k}/N_{v\mathbf{c}}$$

D	I	S	G	L
0	1	1	1	1
1	1	1	0	0
1	1	0	0	1
1	0	0	0	0
1	1	0	0	1

★ In particular,

$$N_{L\langle 0\rangle 0} = 2 \qquad N_{L\langle 0\rangle 1} = 2$$

$$\theta_{L\langle 0\rangle 0} = \theta_{L\langle 0\rangle 1} = 1/2$$

THIS LECTURE

- DGM semantics
- HMMs
  - DP briefly forward, backward etc.
  - Sampling
- Tree DGM marginalization

WHEN IS FACTORIZATION POSSIBLE?

$$p(x_1, \ldots, x_N) = \prod_{n=1}^{N} p(x_n | \boldsymbol{x}_{\mathrm{pa}(x_n)})$$

p can be factorized over G if it can be expressed as above

INDEPENDENCE I-MAP

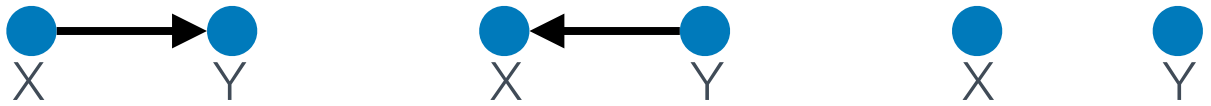
- I(G) (conditional) independences implied by G (not yet defined)
- I(P) (conditional) independences in the distribution P
- G I-map for P in I(G) ⊆ I(P)

p

X	Y	$P(X,Y)$
x^0	y^0	0.08
x^0	y^1	0.32
x^1	y^0	0.12
x^1	y^1	0.48

q

X	Y	$P(X,Y)$
x^0	y^0	0.4
x^0	y^1	0.3
x^1	y^0	0.2
x^1	y^1	0.1



INDEPENDENCE I-MAP

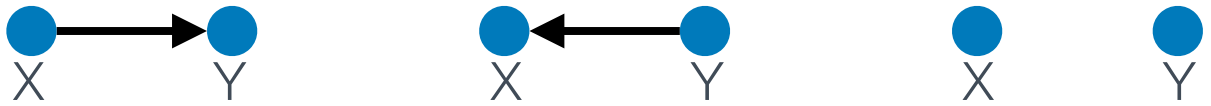
- I(G) independences implied by G (not yet defined)
- I(P) independences in the distribution P
- G I-map for P in I(G) ⊆ I(P)

p

X	Y	$P(X,Y)$
x^0	y^0	0.08
x^0	y^1	0.32
x^1	y^0	0.12
x^1	y^1	0.48

q

X	Y	$P(X,Y)$
x^0	y^0	0.4
x^0	y^1	0.3
x^1	y^0	0.2
x^1	y^1	0.1



- p: X and Y ind. ex. p(X=1) = 0.48 + 0.12 =0.6, p(Y=1) = 0.8, and p(X=1,Y=1) = 0.48
- q: X and Y are dependent

INDEPENDENCE I-MAP

- I(G) independences implied by G (not yet defined)
- I(P) independences in the distribution P
- G I-map for P in I(G) ⊆ I(P)

p

X	Y	$P(X,Y)$
x^0	y^0	0.08
x^0	y^1	0.32
x^1	y^0	0.12
x^1	y^1	0.48

q

X	Y	$P(X,Y)$
x^0	y^0	0.4
x^0	y^1	0.3
x^1	y^0	0.2
x^1	y^1	0.1

G_1

G_2

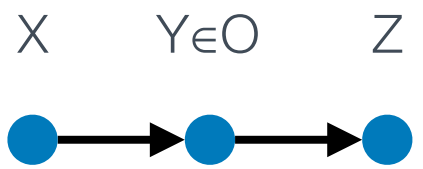
G_3

- All three graphs are I-maps for p
- G_1 and G_2 are I-maps for q, but G_3 is not

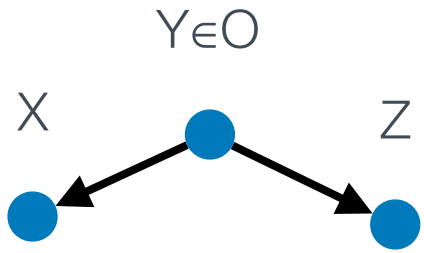
D-SEPARATION

★ A path is d-separated by O if it has

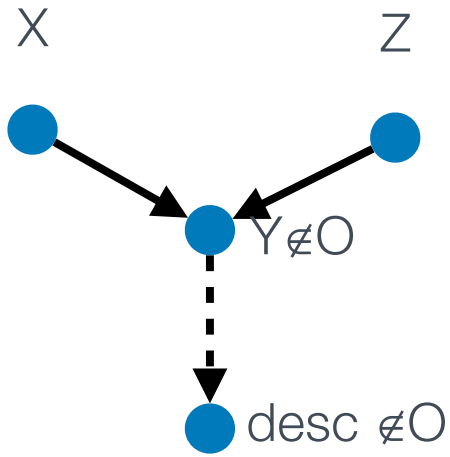
- a chain $X \rightarrow Y \rightarrow Z$ where $Y \in O$
- a fork $X \leftarrow Y \rightarrow Z$ where $Y \in O$
- a v-structure $X \rightarrow Y \leftarrow Z$ where $(Y \cup \text{desc}(Y)) \cap O = \varnothing$



Chain

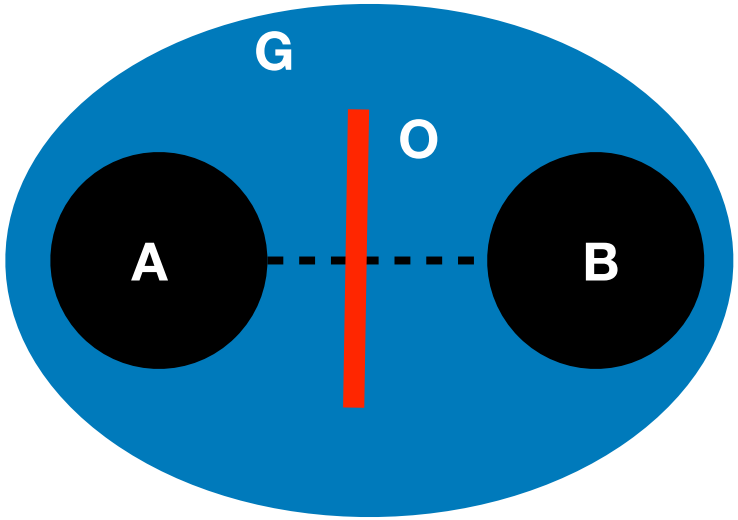


Fork



v-struct

D-SEPARATION SETS AND CI OF DAGS



★ A is d-separated from B given O if every undirected path between A and B is d-separated by O

★ Cond. ind rel. in DAG G,

$$\boldsymbol{x}_A \perp_G \boldsymbol{x}_B | \boldsymbol{x}_O$$

A is d-separated from B given O

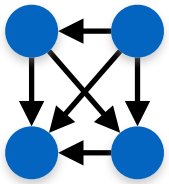
SOUNDNESS AND COMPLETENESS

- I(G) conditional independence relations implied by d-sep in G
- I(p) conditional independence relations satisfied by p

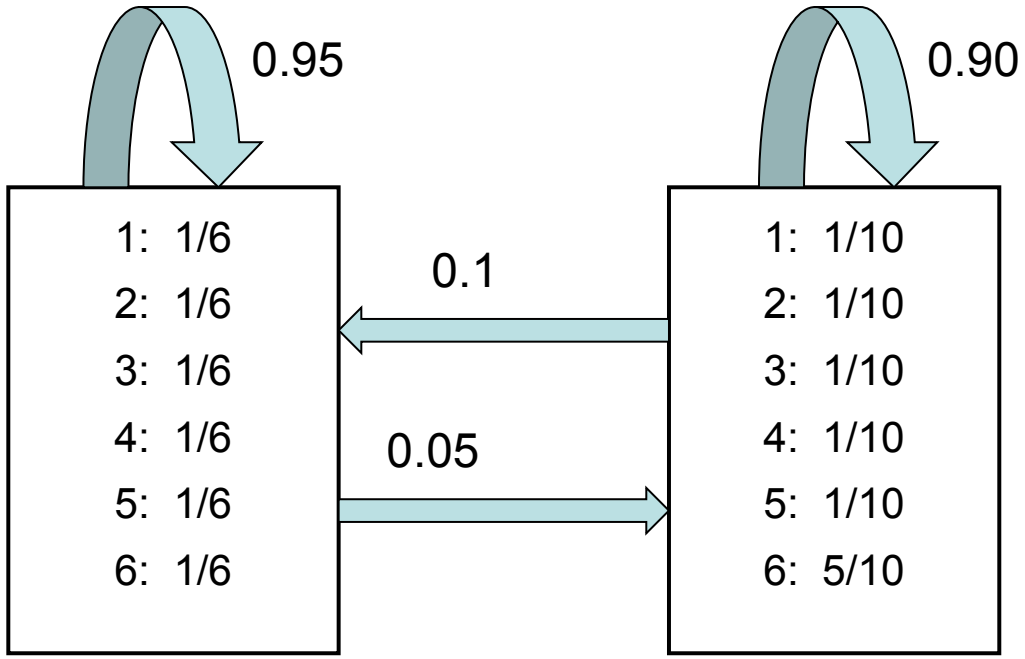
- Theorem

  A distribution P can be factorised over G iff $I(G) \subseteq I(p)$

- "=" not possible to achieve, ex. clique and independent distribution



EMISSION & TRANSTITION DISTRIBUTIONS



THE JOINT DISTRIBUTION

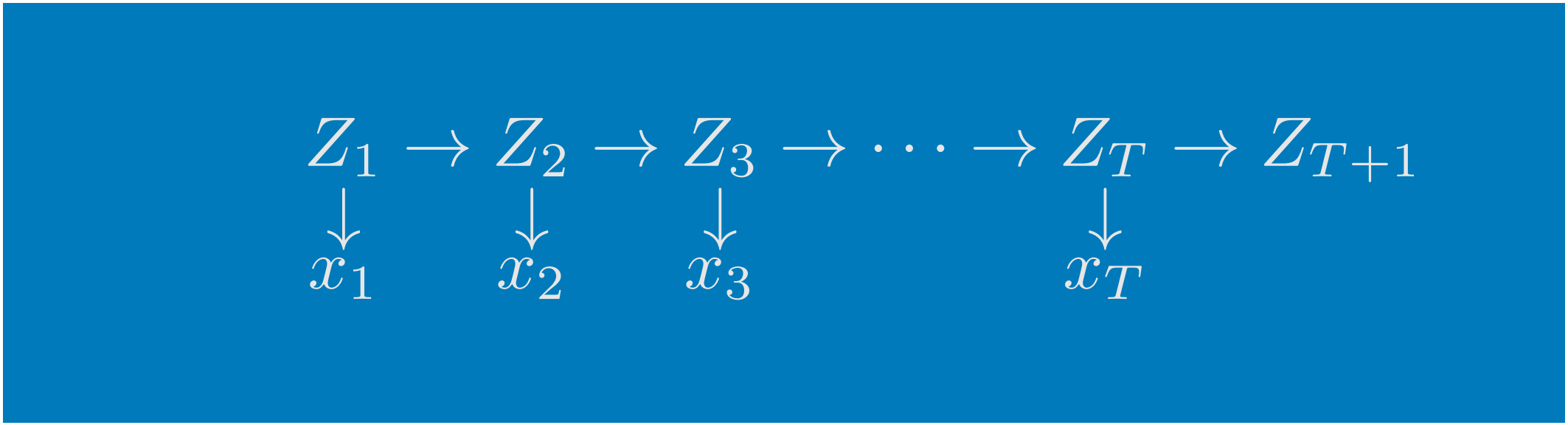
$$
\begin{aligned}
p(\boldsymbol{x}_{1:T}, \boldsymbol{z}_{1:T+1}) &= p(\boldsymbol{x}_{1:T}|\boldsymbol{z}_{1:T})p(\boldsymbol{z}_{1:T+1}) \\
&= p(\boldsymbol{z}_1)\left(\prod_{t=1}^{T} p(\boldsymbol{z}_{t+1}|\boldsymbol{z}_t)\right)\left(\prod_{t=1}^{T} p(\boldsymbol{x}_t|\boldsymbol{z}_t)\right)
\end{aligned}
$$

Categorial (or Gaussian)

- ★ Starts in the state $z_1=k^*$
- ★ When in state z_t
  - emits $p(x_t|z_t)$ B_{x_t,z_t}
  - transits to $p(z_{t+1}|z_t)$ A_{z_{t+1},z_t}
- ★ Stops when reaching a stop state
- ★ z_{T+1} is assumed to be stop

The parameters Now given!!

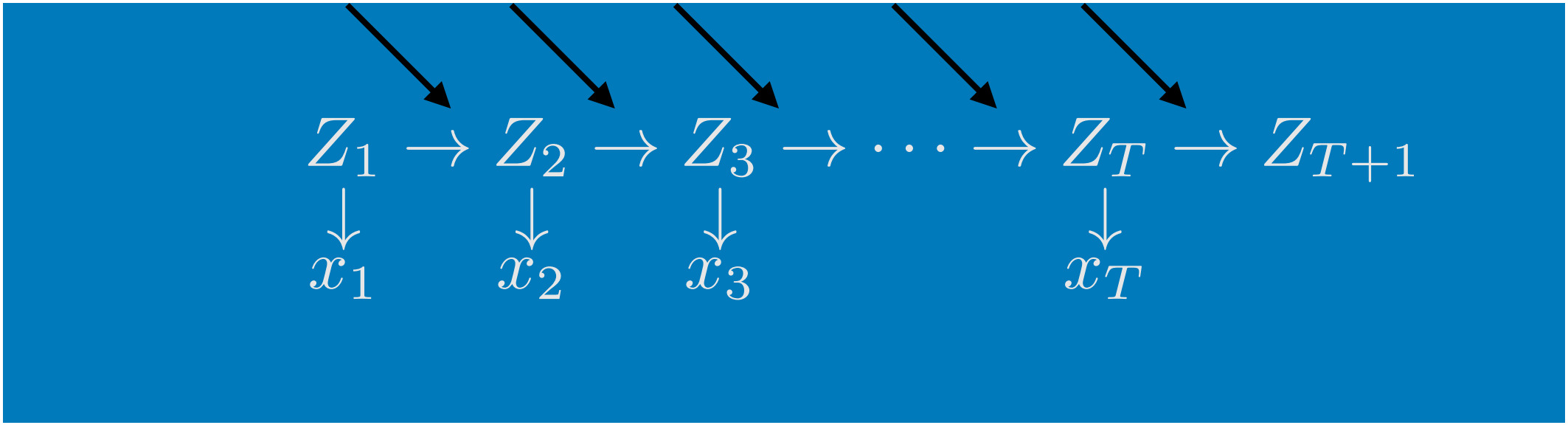
AN HMM CAN BE SEEN AS A DGM



- Z_i hidden
- X_i observable
- Hidden often not observable when training, never when applying

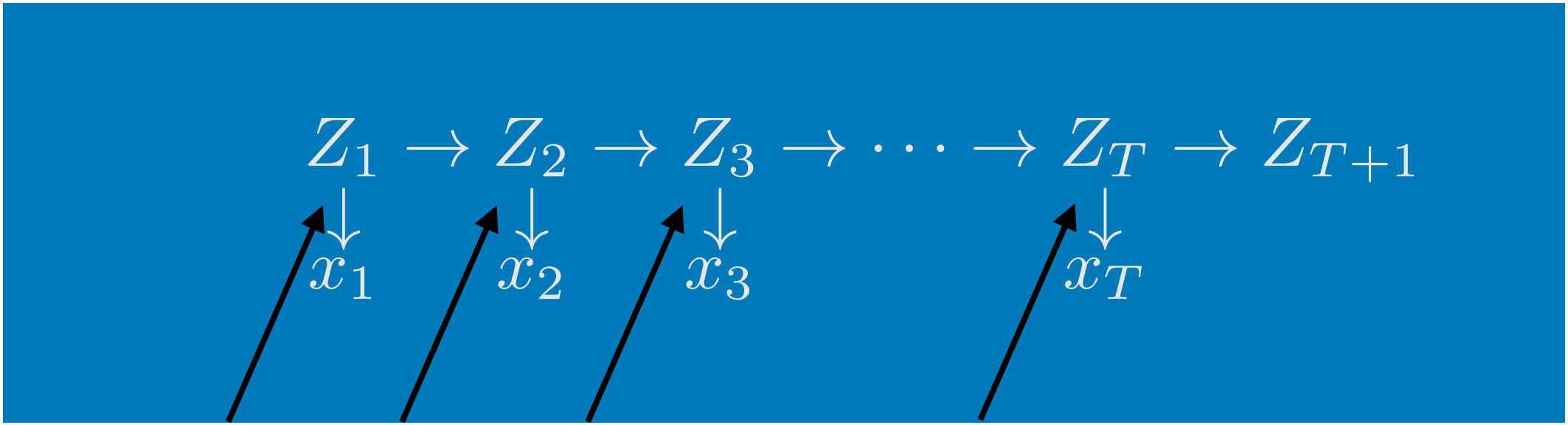
All the same

State	1	2	3	4
1	A_{11}	A_{21}	A_{31}	A_{41}
2	A_{12}	A_{22}	A_{32}	A_{42}
3	A_{13}	A_{23}	A_{33}	A_{43}
4	A_{14}	A_{24}	A_{34}	A_{44}



TRANSITION PROBABILITIES FOR 4 STATES HMM

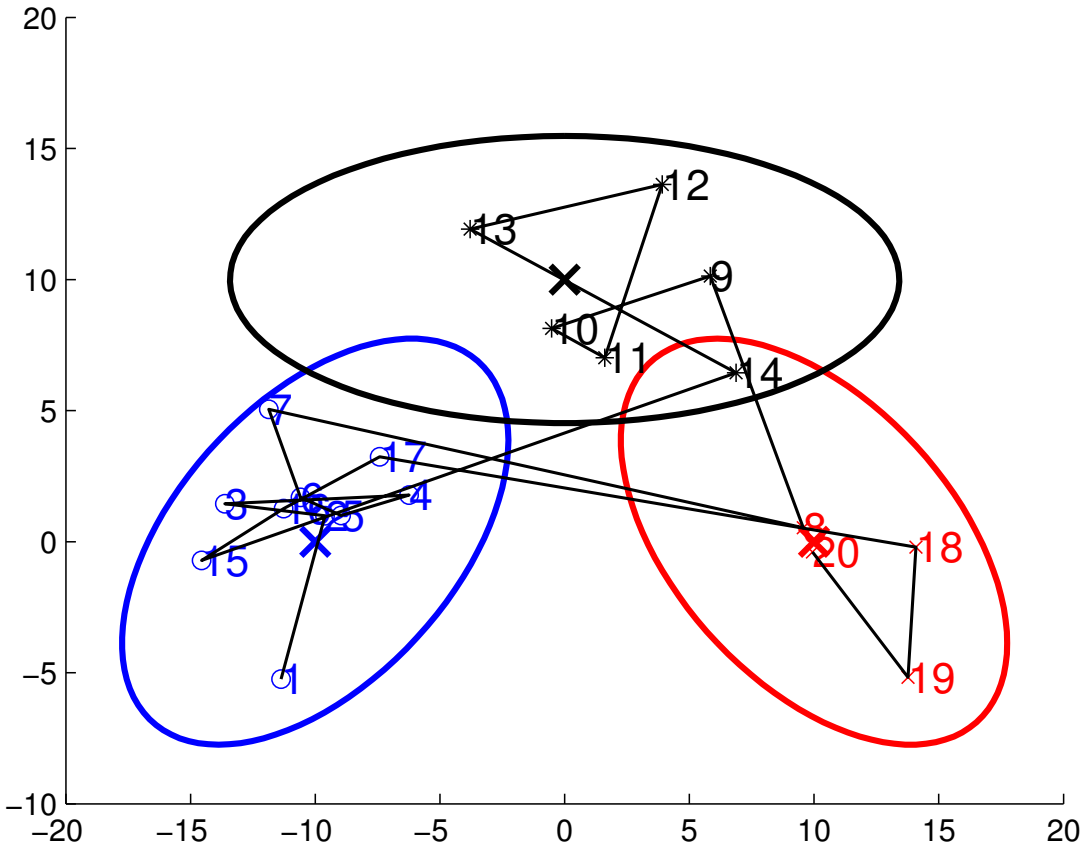
EMISSION PROBABILITIES - HMM WITH 4 STATES & 3 SYMBOLS

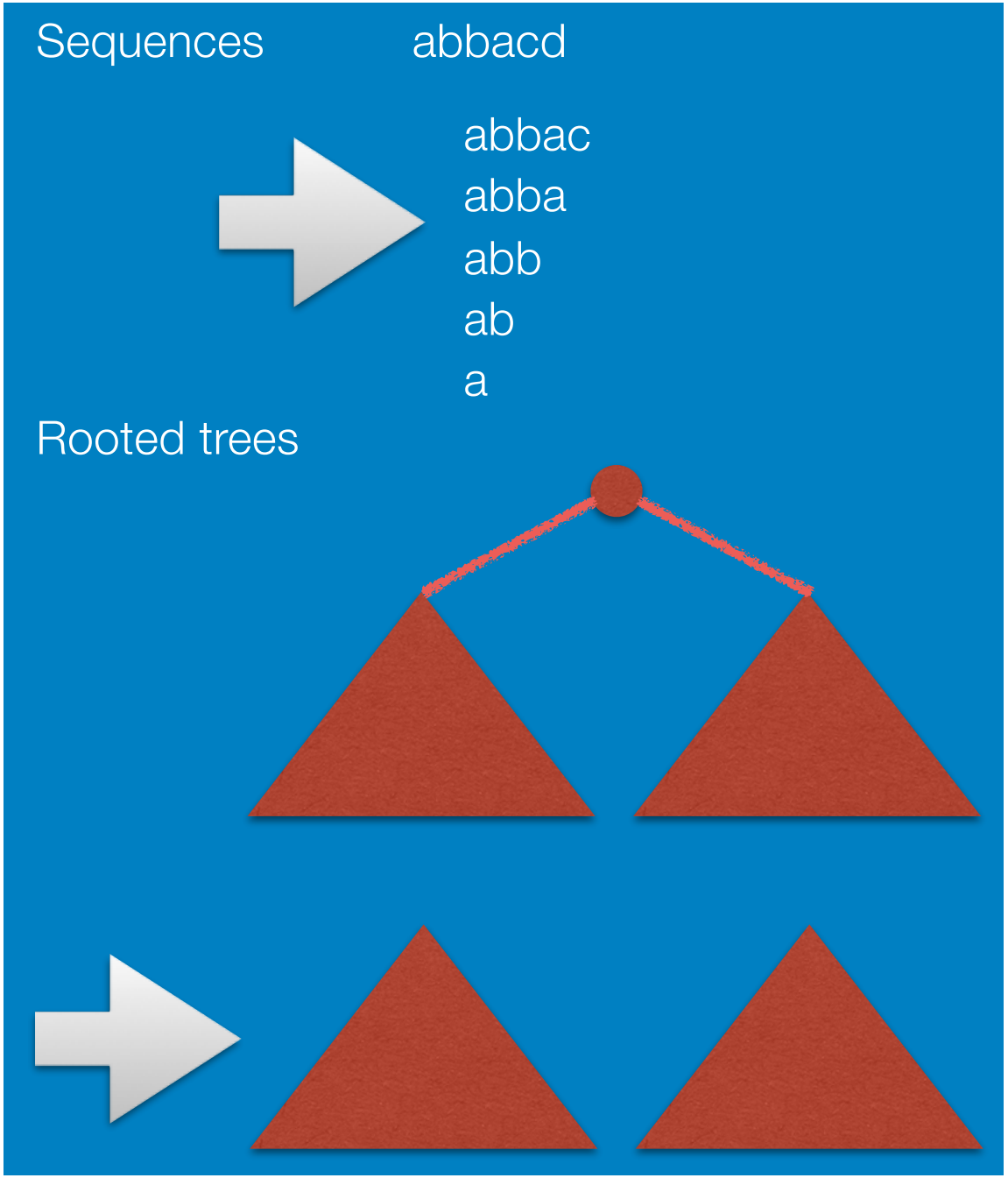


All the same

State\Symb	1	2	3
1	B_{11}	B_{21}	B_{31}
2	B_{12}	B_{22}	B_{32}
3	B_{13}	B_{23}	B_{33}
4	B_{14}	B_{24}	B_{34}

GAUSSIAN EMISSIONS AND HIDDEN STATES





DP

- What is a subproblem?
- What is a subsolution?
- How do we decompose into smaller subproblems?
- How do we combine subsolutions into larger?
- How do we enumerate?
- How many and what time?

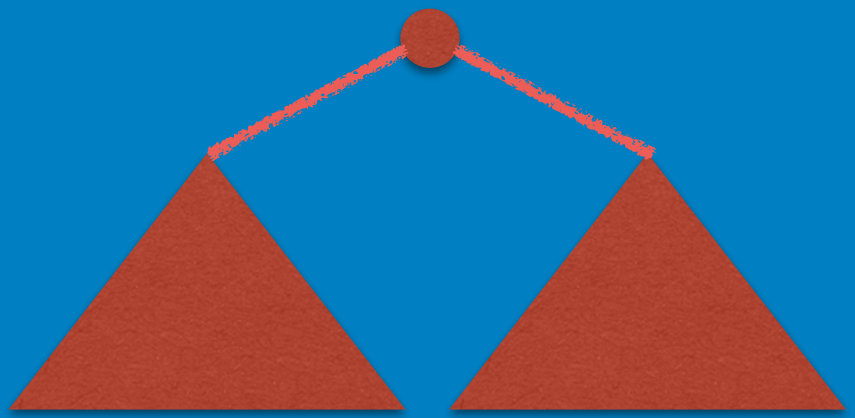
DP

Polynomial many	• What is a subproblem?
	• What is a subsolution?
Polynomial time	• How do we decompose into smaller subproblems?
Polynomial time	• How do we combine subsolutions into larger?
Polynomial time	• How do we enumerate?
Polynomial time overall	• How many and what time?

Sequences

$$Z_1 \rightarrow Z_2 \rightarrow Z_3$$
$$\downarrow \quad \downarrow \quad \downarrow$$
$$x_1 \quad x_2 \quad x_3$$

Rooted trees



DP, SUM RULE, & CONDITIONING

- How do we decompose into smaller subproblems?

Dynamic programming is similar to divide + conquer in that it solves a problem by dividing it into sub-problems. However, in the dynamic programming paradigm, the larger problem is solved by solving and remembering overlapping sub-problems, which are reused repeatedly in the process.

$$P(A) = \sum_i P(A, B = i)$$

Applying sum rule

Notice, by the sum rule,

$$f_t(k) = p(x_{1:t-1}, Z_t = k) = \sum_{k' \in [K]} p(x_{1:t-1}, Z_{t-1} = k', Z_t = k)$$

The set of states

each term in the sum is a probability of an event

$$\begin{array}{ccccccccccccc} ? & \to & ? & \to & ? & \cdots & Z_{t-1} = k' & \to & Z_t = k & \to ? \cdots \\ \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow & \\ x_1 & & x_2 & & x_3 & & x_{t-1} & & ? & \end{array}$$

which, as noted, can be broken into smaller

Applying sum rule

Notice, by the sum rule,

$$f_t(k) = p(x_{1:t-1}, Z_t = k) = \sum_{k' \in [K]} p(x_{1:t-1}, Z_{t-1} = k', Z_t = k)$$

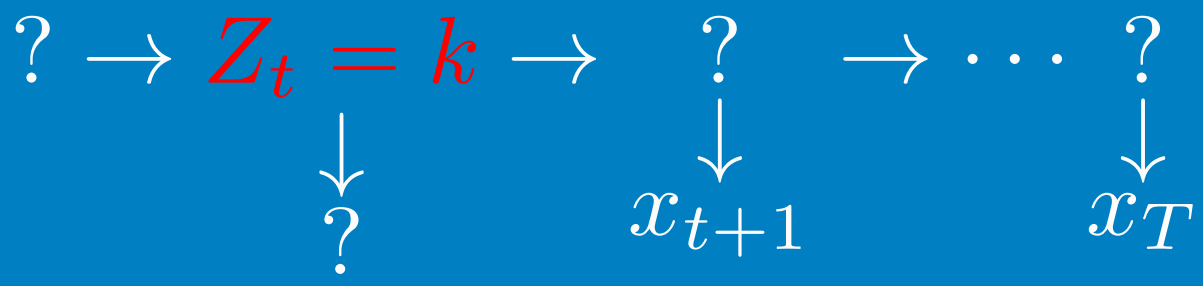
$$= \sum_{k'} \underbrace{f_{t-1}(k')}_{\text{smaller}} \underbrace{p(\boldsymbol{x}_{t-1} | \boldsymbol{Z}_{t-1} = k')}_{\text{emission}} \underbrace{p(\boldsymbol{Z}_t = k | \boldsymbol{Z}_{t-1} = k')}_{\text{transition}}$$

Backward variable

Defined by

$$b_t(k) := p(\boldsymbol{x}_{t+1:T} | \boldsymbol{Z}_t = k)$$

"Graphical model"



Backward recursion

$$b_t(k) := p(\boldsymbol{x}_{t+1:T} | \boldsymbol{Z}_t = k)$$

- DP also for the backward variable b_t

$$b_t(k) = \sum_l \underbrace{p(\boldsymbol{Z}_{t+1} = l | \boldsymbol{Z}_t = k)}_{\text{transition}} \underbrace{b_{t+1}(l)}_{\text{"smaller"}} \underbrace{p(\boldsymbol{x}_{t+1} | \boldsymbol{Z}_{t+1} = l)}_{\text{emission}}$$

ALGORITHM - MARGINALIZATION TREE DGM

- Given DGM with
  - G=T binary rooted directed tree with vertex set V
    all edges are directed downwards
  - Bernoulli CPDs
  - Observation x_O, O
  - Now O is the leaf set (can be any subset of V)
  - Vertices strictly below the vertex u

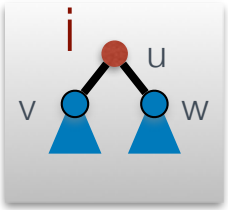
$$u\downarrow = V(T_u)\backslash\{u\}$$

BELOW

- Given DGM with
  - G=T binary rooted directed tree with vertex set V
  - Bernoulli CPDs
  - Observation x_O, where O is the leaf set
- Compute

$$p(x_O)$$

- Subproblems, subsolutions



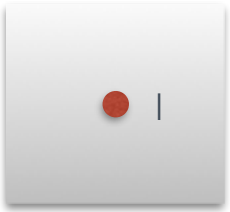
$$s(u,i) = P(x_{\downarrow u \cap O} \mid X_u = i) = \sum_{x_{\downarrow u \setminus O}} P(x_{\downarrow u} \mid X_u = i)$$

THE FINAL ANSWER AT ROOT R

$$
\begin{aligned}
P(x_{\downarrow r \cap O}) &= \sum_i P(x_{\downarrow r \cap O}, X_r = i) \\
&= \sum_i P(x_{\downarrow r \cap O} \mid X_r = i) P(X_r = i) \\
&= \sum_i s(r, i) P(X_r = i)
\end{aligned}
$$

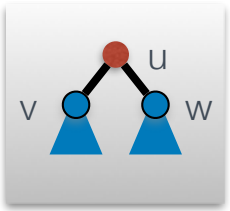
ALGORITHM - MARGINALIZATION TREE DGM

- Visit the vertices of T from leaves to root
  - when at leaf l



$$s(l,i) = \begin{cases} 0 \text{ if } x_l \neq i \\ 1 \text{ if } x_l \neq i \end{cases}$$

  - when at vertex u with children v and w

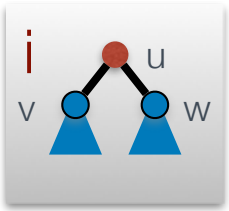


$$s(u,i) = \left(\sum_{j \in \{0,1\}} P(X_v = j \mid X_u = i) s(v,j) \right) \left(\sum_{j \in \{0,1\}} P(X_w = j \mid X_u = i) s(w,j) \right)$$

CPD for uv — Smaller — CPD for uw — Smaller

ALGORITHM - MARGINALIZATION TREE DGM

- Definition vs computation

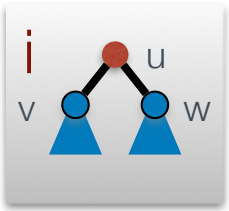


$$s(u,i) = P(x_{\downarrow u \cap O} \,|\, X_u = i)$$

$$? \quad s(u,i) = \left(\sum_{j \in \{0,1\}} P(X_v = j \,|\, X_u = i) s(v,j) \right) \left(\sum_{j \in \{0,1\}} P(X_w = j \,|\, X_u = i) s(w,j) \right)$$

ALGORITHM - MARGINALIZATION TREE DGM

- Definition vs computation

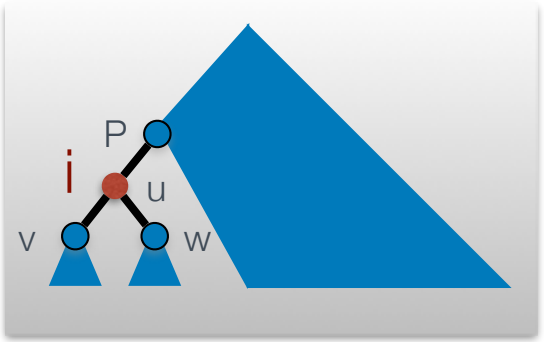


$$s(u,i) = P(x_{\downarrow u \cap O} \,|\, X_u = i)$$

$$s(u,i) = \left(\sum_{j \in \{0,1\}} P(X_v = j \,|\, X_u = i) s(v,j) \right) \left(\sum_{j \in \{0,1\}} P(X_w = j \,|\, X_u = i) s(w,j) \right)$$

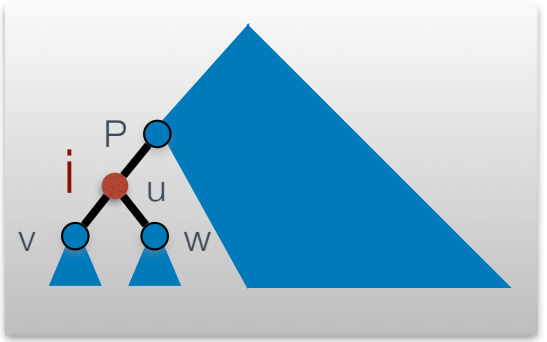
THE MARGINAL

$$P(X_u = i \mid x_O)$$



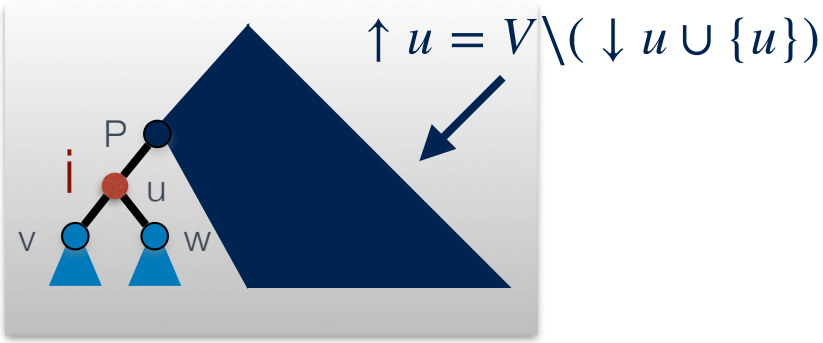
THE MARGINAL OF TREE DGM

$$P(x_u = i \,|\, x_O) \propto P(x_u = i, x_O)$$



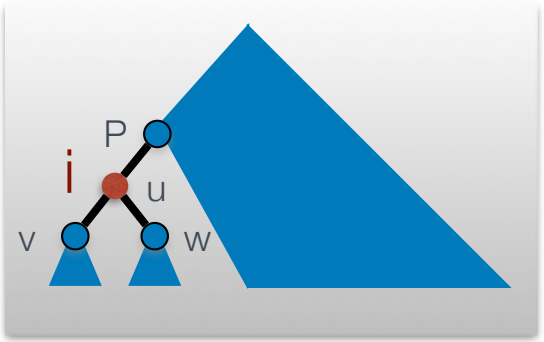
THE MARGINAL

$$P(X_u = i \mid x_O) \propto P(X_u = i, x_O)$$



AGAIN THE LEAF SET

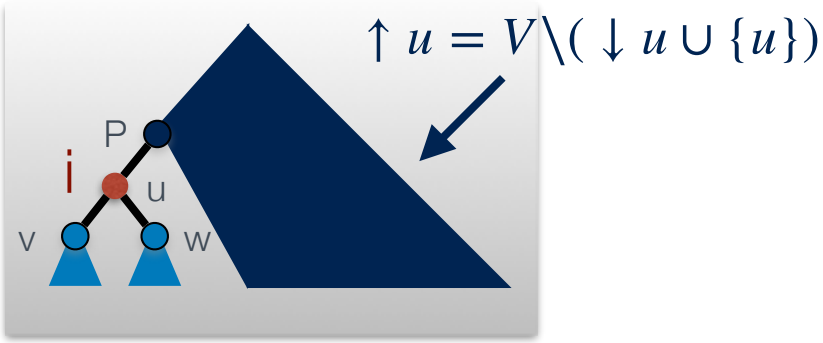
$$P(X_u = i, x_O)$$



- Assume that O is the leaf set

THE REST

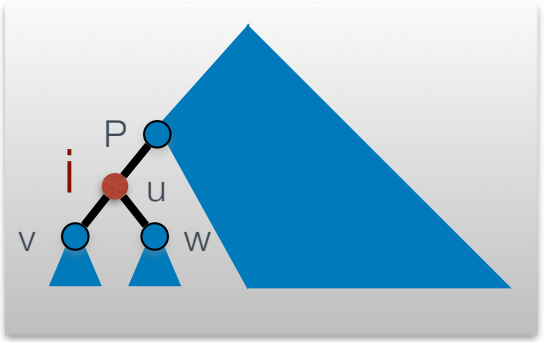
$$P(X_u = i, x_O)$$



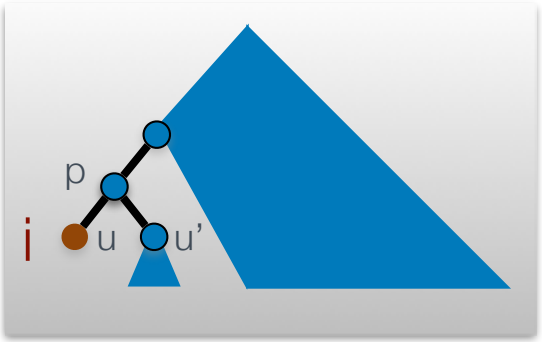
EXPRESSING THE JOINT

$$P(X_u = i, x_O)$$

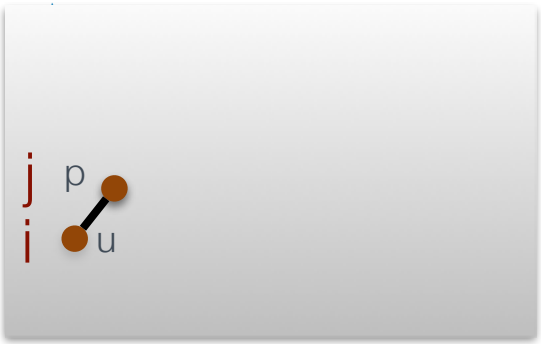
$$= P(x_{\downarrow u \cap O} \mid X_u = i) P(x_{\uparrow u \cap O}, X_u = i)$$

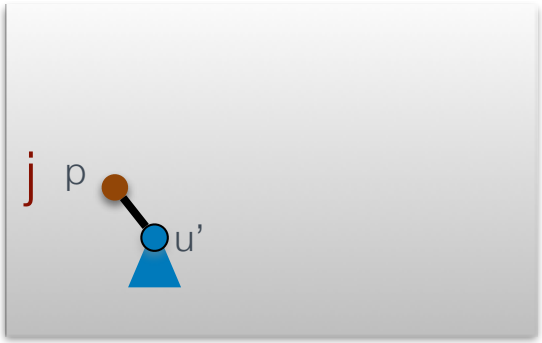


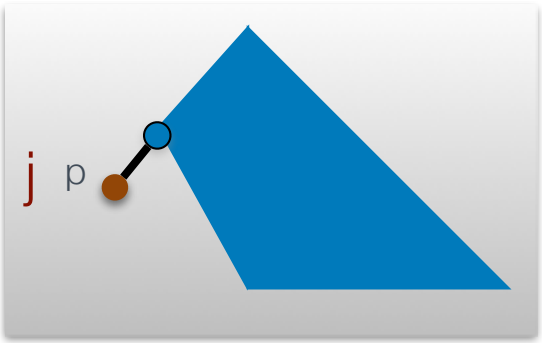
THE REST



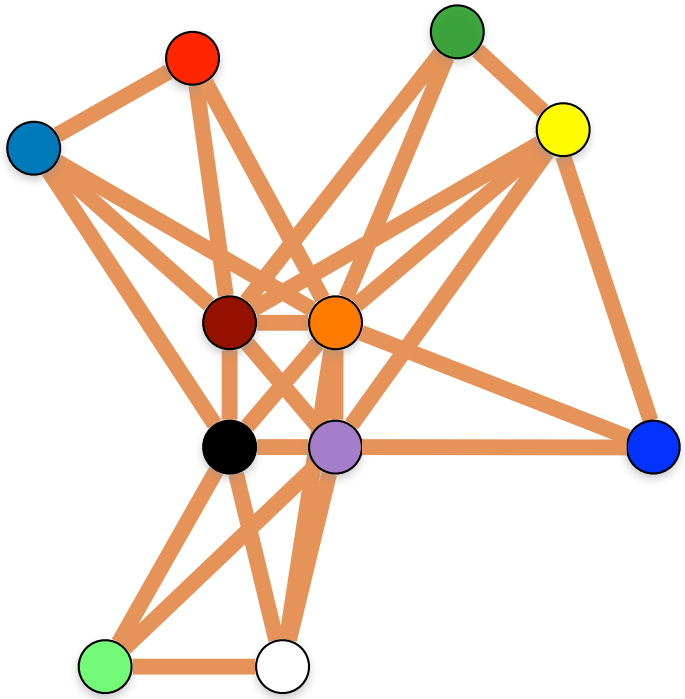
$$t(u,i) = P(x_{\uparrow u \cap O}, X_u = i)$$

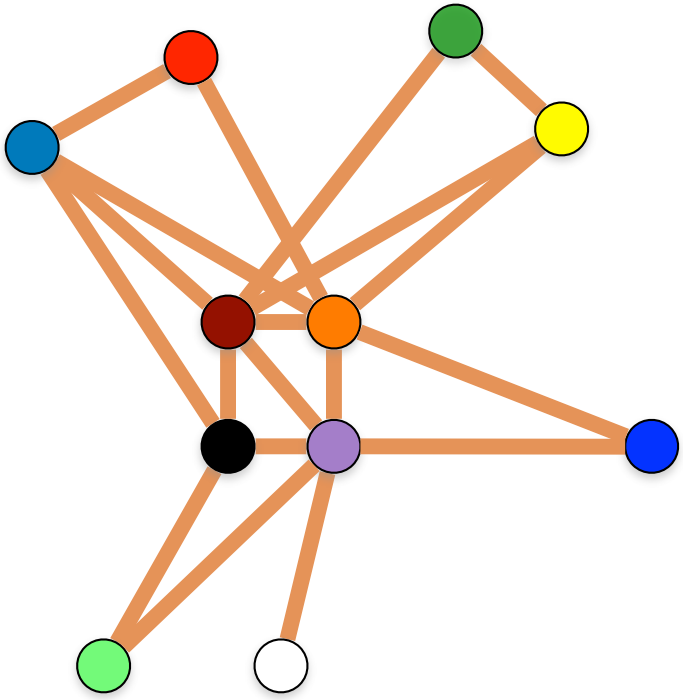






3-TREE & PARTIAL 3-TREE





THE END